Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Original) A method comprising:

executing corresponding instruction threads as a leading thread and a trailing thread;

saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register;

comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread; and

restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault.

2. (Original) The method of claim 1 wherein the leading thread and the trailing thread are executed by a single processor.

3. (Original) The method of claim 1 wherein the leading thread and the trailing thread are executed by multiple processors.

4. (Original) The method of claim 1 wherein the processor state is stored in an entry in the history buffer that stores an instruction pointer to the selected instruction, a value stored in the destination register, wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a register map that indicates a mapping of one or more architectural registers to one or more physical registers.

5. (Original) The method of claim 1 wherein restoring the processor state corresponding to a previous instruction if the comparison indicates a fault comprises:

selectively flushing results of instructions that started execution after an instruction causing the fault started execution;

restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer.

6. (Original) The method of claim 5 wherein selectively flushing results of instructions that started execution after an instruction causing the fault started execution comprises:

flushing non-retired speculative instructions from the leading thread execution circuitry;

flushing an architectural state of the trailing thread from the trailing thread execution circuitry; and

flushing the history buffer after register values used to restore the architectural state to the checkpoint are retrieved.

7. (Original) The method of claim 5 wherein the checkpoint corresponds to the architectural state at a time at which execution of the instruction causing the fault is started.

8. (Original) The method of claim 5 wherein the checkpoint corresponds to the architectural state at a time prior to which execution of the instruction causing the fault is started.

9. (Original) The method of claim 5 wherein restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from a history buffer comprises:

retrieving previous register values for registers written by or subsequent to the instruction causing the fault from the history buffer for the leading thread;

restoring the previous register values to provide a restored architectural state for the leading thread; and

copying the restored architectural state for the leading thread to an architectural register file for the trailing thread to provide a restored architectural state for the trailing thread.

10. (Original) The method of claim 1 wherein the selected instruction comprises a branch instruction.

11. (Original) An apparatus comprising:

means for executing corresponding threads as a leading thread and a trailing thread;

means for saving a processor state corresponding to execution of a selected instruction before writing a result from the selected instruction to a destination register;

means for comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread; and

means for restoring the processor state corresponding to a previous instruction if the comparison indicates a fault.

12. (Original) The apparatus of claim 11 wherein the means for restoring the processor state corresponding to a previous instruction if the comparison indicates a fault comprises:

means for selectively flushing results of instructions that started execution after an instruction causing the fault started execution;

means for restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from a history buffer.

13. (Original) The method of claim 11 wherein the means for selectively flushing results of instructions that started execution after an instruction causing the fault started execution comprises:

means for flushing non-retired speculative instructions from a thread having the instruction that caused the fault;

means for flushing an architectural state of a trailing thread having the instruction that caused the fault; and

means for flushing a history buffer after register values used to restore the architectural state to the checkpoint are retrieved.

14. (Original) An apparatus comprising:

leading thread execution circuitry to execute a leading thread of instructions;

trailing thread execution circuitry to execute a trailing thread of instructions; and

a history buffer coupled with the leading thread execution circuitry and the trailing thread execution circuitry to store information related to execution of a selected instruction from the leading thread of instructions, wherein the information stored in the history buffer is used to restore an architectural state corresponding to a checkpoint if an execution fault is detected.

15. (Original) The apparatus of claim 14 wherein the history buffer stores an instruction pointer to the selected instruction, a value stored in the destination register, wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a register map that indicates a mapping of architectural registers to physical registers.

16. (Original) The apparatus of claim 15 wherein the architectural state corresponding to the checkpoint is restored by selectively flushing results of instructions that started execution after an instruction causing the fault started execution and restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer.

17. (Original) The apparatus of claim 16 wherein the architectural state corresponding to the checkpoint is restored by flushing non-retired speculative instructions from the execution circuitry corresponding to the thread having an instruction that caused the fault, flushing an architectural state of the execution circuitry corresponding to the

thread having the instruction that caused the fault, and flushing the history buffer after register values used to restore the architectural state to the checkpoint are retrieved.

18. (Original) The apparatus of claim 14 wherein the execution fault is caused by a branch instruction.

19. (Original) The apparatus of claim 14 wherein the checkpoint corresponds to the architectural state at a time at which an instruction causing the fault is started.

20. (Original) The apparatus of claim 14 wherein the checkpoint corresponds to the architectural state at a time prior to which an instruction causing the fault is started.

21. (Original) The apparatus of claim 14 wherein the leading thread execution circuitry and the trailing thread execution circuitry are part of a single processor.

22. (Original) The apparatus of claim 14 wherein the leading thread execution circuitry is part of a first processor and the trailing thread execution circuitry are part of a second processor.

23. (Original) A system comprising:

leading thread execution circuitry to execute a leading thread of instructions;

trailing thread execution circuitry to execute a trailing thread of instructions;

an input/output controller coupled with the leading thread execution circuitry; and

a history buffer coupled with the leading thread execution circuitry and the trailing thread execution circuitry to store information related execution of a selected instruction from the leading thread of instructions, wherein the information stored in the history buffer is used to restore an architectural state corresponding to a checkpoint if an execution fault is detected.

24. (Original) The system of claim 23 wherein the history buffer stores an instruction pointer to the selected instruction, a value stored in the destination register,

wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a register map that indicates a mapping of architectural registers to physical registers.

25. (Original) The system of claim 24 wherein the architectural state corresponding to the checkpoint is restored by selectively flushing results of instructions that started execution after an instruction causing the fault started execution and restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer.

26. (Original) The system of claim 25 wherein the architectural state corresponding to the checkpoint is restored by flushing non-retired speculative instructions from the execution circuitry corresponding to the thread having an instruction that caused the fault, flushing an architectural state of the execution circuitry corresponding to the thread having the instruction that caused the fault, and flushing the history buffer after register values used to restore the architectural state to the checkpoint are retrieved.

27. (Original) The system of claim 23 wherein the execution fault is caused by a branch instruction.

28. (Original) The system of claim 23 wherein the checkpoint corresponds to the architectural state at a time at which an instruction causing the fault is started.

29. (Original) The system of claim 23 wherein the leading thread execution circuitry is part of a first processor and the trailing thread execution circuitry are part of a second processor.